This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (Currently Amended) A system for automatically generating procedural-oriented output source code from a functional model comprising:

a computer display;

a modeler for defining at least one of a plurality of code elements and a structure of a code block and generating a graphical representation on said computer display of the at least one code element and structure of the code block, wherein the modeler processes input comprising a code block of procedural-oriented source code from an innermost element to an outermost element and generates from the input a ~~code~~ functional model comprising a graphical representation of a structure and flow of the code block; ~~and~~

a selector for selecting at least one of a plurality of programming languages in which to generate procedural-oriented output source code from the functional model; and

generating procedural-oriented output source code from the functional model.

2. (Previously presented) The system of claim 1, further comprising a user interface for receiving the definition of the at least one code element and the structure of the code block.

3. (Canceled)

4. (Previously presented) The system of claim 1, further comprising a code generator for receiving the graphical representation of the at least one code element and the structure of the code block and the at least one programming language and generating procedural-oriented output source code in each of the at least one programming languages.

5.– 7. (Canceled)

8. (Currently Amended) A method of automatically generating procedural-oriented output source code from a functional software model comprising:

processing a block of procedural-oriented programming code from an innermost element to an outermost element and generating from the processed block of procedural-oriented programming code a functional software model;

defining a plurality of code elements within the block of procedural-oriented programming code;

specifying a structure of the block of procedural-oriented programming code including the plurality of code elements; ~~and~~

generating from the plurality of code elements and the structure of the block of procedural-oriented programming code including the plurality of code elements a functional software model, wherein the functional software model comprises a graphical representation of the plurality of code elements and flow of the block of procedural-oriented programming code; ~~and~~

specifying at least one target language in which procedural-oriented output source code for the graphical representation is to be generated, wherein the at least one target language specified is different from a language of the processed block of procedural-oriented programming code; and

generating procedural-oriented output source code in the at least one target language from the functional model.

9. (Previously presented) The method of claim 8, further comprising receiving the definition of the plurality of code elements with the block of procedural-oriented programming code and specifying the structure of the block of programming code via a user interface.

10. (Canceled).

11. (Currently Amended) The method of claim ~~10~~ 8, further comprising generating the procedural-oriented output source code in the at least one target language.

12. (Previously presented) The method of claim 8, wherein one of the plurality of code elements comprises a variable, comment, constant, object, function, method, prototype, member, data type, callback, delegate, reference, field, variant, property, interface, class, type, enumeration, structure, primitive, array, or event handle.

13. (Previously presented) The method of claim 8, wherein one of the plurality of code elements comprises a code relation.

14. (Original) The method of claim 13, wherein the code relation comprises a mathematical operator.

15. (Previously presented) The method of claim 8, wherein one of the plurality of code elements comprises an evaluation entity.

16. (Previously presented) The method of claim 15, wherein the evaluation entity comprises one of a method call, a plurality of code entities, a plurality of code relations, or an instantiation of a class.

17. (Previously presented) The method of claim 8, wherein one of the plurality of code elements comprises a passive entity.

18. (Previously presented) The method of claim 17, wherein the passive entity comprises a comment or a modeling diagram.

19. (Previously presented) The method of claim 8, wherein one of the plurality of code elements comprises a block entity.

20. (Previously presented) The method of claim 19, wherein the block entity comprises a method entity, a member entity, a class entity, a namespace entity, or a file entity.

21. (Original) The method of claim 20, wherein a many-to-many relationship exists between block entities.

22. (Currently Amended) A computer-readable storage medium including computer-readable instructions for performing a method comprising:

processing a block of procedural-oriented programming code from an innermost program element to an outermost program element and generating from the processed block of procedural-oriented programming code a functional software model;

defining a plurality of code elements within the block of procedural-oriented programming code;

specifying a structure of the block of procedural-oriented programming code including the plurality of code elements; ~~and~~

generating from the plurality of code elements and the structure of the block of procedural-oriented programming code including the plurality of code elements a functional software model, wherein the functional software model comprises a graphical representation of the plurality of code elements and flow of the block of procedural-oriented programming code comprising the functional software model; and

generating procedural-oriented output source code from the functional model.